



APPLICATION NO. 09/846,410

TITLE OF INVENTION: Multiple Data Rate Hybrid Walsh Codes for
CDMA

INVENTOR: Urbain A. von der Embse

Currently amended Claims

APPLICATION NO. 09/846.410

TITLE OF INVENTION: Multiple Data Rate Hybrid Walsh Codes
for CDMA

5 INVENTORS: Urbain A. von der Embse

CLAIMS

10

WHAT IS CLAIMED IS:

Claim 1. (cancelled)

Claim 2. (cancelled)

Claim 3. (cancelled)

15

Claim 4. (cancelled)

20

Claim 5. (currently amended) A method for generating and applying hybrid Walsh complex orthogonal codes for code division multiple access (CDMA), said method comprising the steps: the implementation of design and implementation of fast encoders and fast decoders for Hybrid Walsh and generalized Hybrid hybrid Walsh complex orthogonal codes for CDMA, channelization codes for multiple data rate users over said method comprising the steps: a frequency band with properties

25

generating N Walsh codes $W(c)$ with code index $c=0,1,2,\dots,N-1$, each with N chips where N is a power of 2,

generating said N hybrid Walsh codes $\tilde{W}(c)$ by re-ordering each of said N Walsh codes into a corresponding real component and a corresponding imaginary component of a hybrid Walsh code as defined by equations

30

$$\begin{aligned} \text{for } c &= 0, & \tilde{W}(c) &= W(0) + jW(0) \\ \text{for } c &= 1, 2, \dots, N/2-1, & \tilde{W}(c) &= W(2c) + jW(2c-1) \end{aligned}$$

~~_____~~ for $c = N/2$, $\tilde{W}(c) = W(N-1) + jW(N-1)$
~~_____~~ for $c = N/2+1, \dots, N-1$, $\tilde{W}(c) = W(2N-2c-1) + jW(2N-2c)$
 wherein $j = \sqrt{-1}$,

wherein said hybrid Walsh codes are generated by reading ~~each~~ the
 5 ~~chip values from said N Walsh codes~~ chip values from a Walsh
code memory in a digital signal processor and writing to
~~said hybrid Walsh memories using a~~ hybrid Walsh code memory
using addresses specified by said re-orderings of said
N Walsh codes,

10 ~~codes and,~~
 applying said hybrid Walsh codes in ~~the~~ an encoder and in ~~the~~ a
decoder of a CDMA system by replacing existing said N Walsh
real codes with said hybrid Walsh complex codes using the
according to a same code vector indexing-, and
 15 transmitting data encoded by the encoder and receiving data
decoded by the decoder.

~~Hybrid Walsh inphase (real axis) codes and quadrature~~
~~(imaginary axis) codes are defined by lexicographic reordering~~
~~permutations of the Walsh code~~

20 ~~Hybrid Walsh codes have a 1 to 1 sequency-frequency~~
~~correspondence with the DFT codes and have a 1 to 1 even-cosine~~
~~and odd-sine correspondences with the DFT codes~~

25 ~~Hybrid Walsh codes take values $\{1+j, 1+j, -1-j, -1-j\}$ or~~
~~equivalently take values $\{1, j, -1, -j\}$ with a (-45) rotation of~~
~~axes and a renormalization~~

Claim 6. (currently amended) A method for generating and
 30 applying spreading codes for code division multiple access
(CDMA), comprising the steps: for the implementation of design
~~and implementation of encoders and decoders for complex~~
~~orthogonal CDMA and hybrid Walsh codes for CDMA as described in~~
~~claim 5, further comprising the steps: complex orthogonal CDMA~~

~~channelization codes for multiple data rate users over a frequency band with properties using tensor products also called Kronecker products to construct a second code,~~

5 ~~constructing a spreading wherein an example 24 chip tensor product code from a 8 chip hybrid Walsh code and a 3 chip discrete Discrete Fourier transform Transform (DFT) code, said spreading 24 chip code is defined by an 24-N*P row by~~

~~24-N*P column code matrix C_{24} wherein row vectors are code vectors and column elements are code chips,~~

10 ~~said 8 chip hybrid Walsh code is defined by a 8-N row by 8 N column code matrix \tilde{W}_8 \tilde{W} ,~~

~~said 3 chip DFT code is defined by a 3-P row by 3-P column code matrix E_3 ,~~

15 ~~said spreading code matrix C_{24} is constructed by tensor a Kronecker product of said hybrid Walsh code matrix \tilde{W} with said DFT code matrix E_3 defined by the equation~~

$$C_{24} = \tilde{W} \otimes E_3$$

~~wherein symbol the operator " \otimes " is a tensor Kronecker product operation,~~

20 ~~row u+1 and column n+1 matrix element $C_{24}(u+1, n+1)$ of said C_{24} is defined by equation~~

$$C_{24}(u+1, n+1) = \tilde{W}_8(u_0+1, n_0+1) E_3(u_1+1, n_1+1)$$

~~wherein~~

$$u = u_0 + 8u_1$$

$$u = 0, 1, \dots, 23$$

$$n = n_0 + 8n_1$$

$$n = 0, 1, \dots, 23$$

25 ~~wherein u, n are code and chip indices for said codes C_{24} and u_0, n_0 are code and chip indices for said code \tilde{W}_8 and u_1, n_1 are code and chip indices for said code E_3 ,~~

30 ~~wherein said encoder and said decoder for CDMA communications have memories assigned to said C_{24} , \tilde{W}_8 , E_3 codes,~~

~~said C_{24} codes are generated by reading code chip values from said \tilde{W}_8 memory and said E_3 memory,~~

~~said chip values are combined using said equations to yield
said chip values stored in said C_{24} memory for said C_{24} ,~~

5 ~~said C_{24} codes are read from said memory and implemented in said
encoder and said decoder,~~

~~using direct products to construct a second code,~~

~~wherein an example 11 chip direct product code is constructed
from said 8 chip hybrid Walsh code and said 3 chip DFT~~

10 ~~code,~~

~~said 11 chip code is defined by the 11 row by 11 column code
matrix C_{11} ,~~

~~said C_{11} is constructed by direct product of said \tilde{W}_8 with said E_3
defined by equation~~

15 ~~$C_{11} = \tilde{W}_8 \oplus E_3$~~

~~wherein symbol " \oplus " is a direct product operation,~~

~~row $u+1$ and column $n+1$ matrix element $C_{11}(u+1, n+1)$ of said C_{11} is
defined by equation~~

~~$C_{11}(u+1, n+1) = \tilde{W}_8(u_0+1, n_0+1)$ for $u=u_0, n=n_0,$~~

20 ~~$= E_3(u_1+1, n_1+1)$ for $u=8+u_1, n=8+n_1,$~~

~~$= 0$ otherwise,~~

~~wherein said encoder and said decoder for CDMA communications~~

~~have memories assigned to said C_{11}, \tilde{W}_8, E_3 codes,~~

~~said C_{11} codes are generated by reading code chip values from said
25 \tilde{W}_8 memory and said E_3 memory and combined using said
equations to yield said chip values for said C_{11} codes and
stored in said C_{11} memory,~~

~~said C_{11} codes are read from memory and implemented in said
encoder and in said decoder,~~

30 ~~using functional combining to construct a second code,~~

~~wherein an example 11 chip functional combined \hat{C}_{11} code is~~

~~constructed from said C_{11} codes by using codes to fill the
two null subspaces of said C_{11} .~~

~~wherein said \hat{C}_{11} codes are read from memory and implemented in~~
~~said encoder and said decoder and,~~
~~using a combinations of tensor products, direct products, and~~
~~functional combining to construct a second code which~~
5 ~~is read from memory and implemented in said encoder~~
~~and said decoder.~~
applying said spreading code matrix C in an encoder and in a
decoder of a CDMA system by replacing existing said N Walsh
real codes with said hybrid Walsh complex codes according
10 to a same code vector indexing, and
transmitting data encoded by the encoder and receiving data
decoded by the decoder.

Claim 7. (currently amended) A method for implementing the
15 ~~implementation of design and implementation of encoders and~~
~~decoders for complex orthogonal CDMA and hybrid Walsh codes for~~
~~CDMA, as described in claim 5, further comprising the steps:~~
~~said encoder operates as a block encoder,~~
~~encoding blocks of received N data symbols with contained in a~~
20 block said with respective N hybrid Walsh codes and
~~summing to yield N chips encoded data symbols for each~~
~~block at the output chip rate of 1/T chips per second~~
~~wherein T is the interval between chips,~~
~~wherein said encoder accepts up to M-N users per block for~~
25 $N=2^M$, wherein N is a power of 2 and M is the actual number
of users represented in the block, each of said users
having a data rate corresponding to one of 1, 2, . . . , N/2
~~said user data symbols per block,~~
~~user data symbols over said block are arranged in packets with~~
30 ~~each packet containing said user data symbols for said~~
~~block,~~
wherein said encoder accepts packets from each user and writes
them to memory "A" for each block, wherein a binary
address index $d=d_0+2d_1+4d_2+\dots+(N/2)d_{M-1}=0, 1, \dots, N-1$ is

comprising a number of bits corresponding to the maximum number of users N is used for addressing of said data symbols stored in memory "A" and the data symbols for each user of the block are stored in memory "A" in a hierarchy such that a particular user is selected according to a number of more significant bits of the binary address index and the data symbols of the particular user are selected according to a number of lesser significant bits of the binary address index, the number of more significant bits and lesser significant bits of the particular user being determined according to the data rate of the particular user and the total number of users M per block.

wherein binary coefficients d_0, d_1, \dots, d_{M-1} take values 0,1,

said binary address index can be independently mapped onto said data symbol addresses of "A" to provide additional flexibility in assigning users to hybrid Walsh vectors, said data symbol address is partitioned into M overlapping algebraic index fields $d_{M-1}, d_{M-2}d_{M-1}, \dots, d_1d_2, \dots, d_{M-2}d_{M-1}, d_0d_1d_2, \dots, d_{M-2}d_{M-1}$, with each field indexed over the allowable number 2,4, ..., $N/2, N$ of said data rate users at symbol rates $1/2T, 1/4T, \dots, 2/NT, 1/NT$ respectively,

assign said users with like data symbol rates to the M groups $u_0, u_1, \dots, u_{M-2}, u_{M-1}$ of users with the respective symbol rates $1/2T, 1/4T, \dots, 2/NT, 1/NT$,

assign said data symbol indices in said index field d_{M-1} to said users in said group u_0 , assign said data symbol indices in said index field $d_{M-2}d_{M-1}$ to said users in said group u_1 , et al and finally assign said data symbol indices in said index field $d_0d_1d_2, \dots, d_{M-2}d_{M-1}$ to said users in said group u_{M-1} use said mapping and assignments to specify said write addresses of said user data symbols onto said input code vector stored in said memory "A" and,

said input vector in said "A" is encoded in said encoder and processed for transmission.

Claim 8. (currently amended) Wherein said hybrid Walsh codes in claim 5 have a fast encoding implementation algorithm, comprising the steps:

- 5 wherein said fast encoding algorithm implemented in the encoder uses memory "A" for input and to support pass 1 and uses memories "B","C" to support passes 2,...,M wherein $N=2^M$ and uses memory "D" to store the encoded chip output from the reordering pass,
- 10 writing input data symbol vector $Z(d_0, d_1, \dots, d_{M-2}, d_{M-1})$ to said "A" wherein the binary addressing word takes address values $d_0 d_1 \dots d_{M-2} d_{M-1} = 0, 1, 2, \dots, N-1,$
- wherein pass 1 reads pairs of data symbols from "A" corresponding to $d_0=0,1$ wherein the addresses of the data symbol for $d_0=0$ are $0, 2, 4, \dots, N-2$ and for $d_0=1$ are $1, 3, \dots, N-1$ and for
- 15 each pair of data symbols pass 1 performs a 2-point hybrid Walsh transform and sums the outputs for each of the encoded chip binary index values $n_{M-1}=0,1$ and writes the outputs to memory "B" using the addresses of the
- 20 respective data symbols corresponding to $d_0=0,1$, and pass 1 processing generates the data vector $Z(n_{M-1}, d_1, \dots, d_{M-1})$ in "B",
- wherein pass $m=2, 3, \dots, N-1$ reads pairs of data symbols from "B","C","B",... and writes the outputs to "C","B","C",..
- 25 ... and data symbol pairs for $d_{m-1}=0,1$ are read over address blocks each of length 2^m and starting with the first address block the data symbol $d_{m-1}=0$ is read for addresses $0, 1, \dots, 2^{(m-1)}$ and for $d_{m-1}=1$ the data symbol addresses are $2^{(m-1)}+1, \dots, 2^m$ and for each pair of data symbols
- 30 pass m performs a 2-point hybrid Walsh transform and sums the outputs for each of the encoded chip binary index values $(n_{M-m+1}, n_{M-m}=0,1)$ and writes the outputs to addresses $(n_{M-m+1}, 0), (n_{M-m+1}, 1)$ corresponding to the addresses of the input $d_{m-1}=0,1$ and this processing is repeated for each of

the address blocks and pass m processing generates the
 data vector $Z(n_{M-1}, \dots, n_{M-m}, d_m, \dots, d_{M-1})$,
 wherein pass M reads pairs of data symbols from "B" or "C" and
 writes the output to "C" or "B" and data symbol pairs for
 5 $d_{M-1}=0,1$ are read over addresses $0,1,\dots,N/2-1$ for $d_{M-1}=0$
 and over addresses $N/2,\dots,M-1$ for $d_{M-1}=1$ and for each
 pair of data symbols pass M performs a 2-point hybrid Walsh
 transform and sums the outputs for each of the encoded chip
 binary index values $(n_1, n_0=0,1)$ and writes the output to
 10 address $(n_1, 0)$, $(n_1, 1)$ corresponding to the address of the
 input $d_{M-1}=0,1$ and pass M processing generates the data
 vector $Z(n_{M-1}, \dots, n_0)$,
 wherein a final reordering pass re-orders the encoded chip
 symbols in memory "B" or "C" and stores the ordered output
 15 $Z(n_0, n_1, \dots, n_{M-2}, n_{M-1})$ in memory "D", and
~~said fast implementation algorithm in encoder uses said memory~~
~~"A" for input and to support pass 1, memories "B", "C" to~~
~~support passes 2, ..., M and re-ordering pass, and memory~~
~~"D" for output,~~
 20 ~~write input data symbol vector $Z(d_0, d_1, \dots, d_{M-2}, d_{M-1})$ to said~~
~~"A" wherein said $(d_0, d_1, \dots, d_{M-2}, d_{M-1})$ is said binary~~
~~addressing index after said mapping of said data vector~~
~~onto said "A",~~
~~pass 1 reads from said "A", performs pass 1, and writes the~~
 25 ~~output to said "B",~~
~~pass 1 multiplies said Z by the kernel $\{(-1)^{dr_0 n_{M-1} + j(-1)^{di_0}}$~~
 ~~$n_{M-1}\}$ and sums over $dr_0, di_0=0,1$ to yield the partially~~
~~encoded symbol set $Z(n_{M-1}, d_1, \dots, d_{M-2}, d_{M-1})$ where $dr_0=er(d_0)$~~
~~and $er(d)$ is the real axis Walsh code for d, $di_0=ei(d_0)$~~
 30 ~~where $ei(d)$ is the imaginary axis Walsh code for d, and n_{M-1}~~
~~is a binary code chip coefficient in said code chip~~
~~indexing $n = n_0 + 2n_1 + \dots + (N/4)n_{M-2} + (N/2)n_{M-1}$~~
~~write said output symbol set $Z(n_{M-1}, d_1, \dots, d_{M-2}, d_{M-1})$ to said~~
~~"B" wherein said address index n_{M-1} replaces said index d_0~~
 35 ~~pass 2 reads from said "B", performs pass 2, and writes the~~

~~output to said "C",~~
~~pass 3 reads from said "C", performs pass 3, and writes the~~
~~output to said "B",~~
~~subsequent passes alternate in read/write from/to said "B" and~~
5 ~~write/read to/from said "C",~~
~~implement passes $m=2,3,\dots,M-1$ of said fast encoding algorithm~~
~~by multiplying~~
 ~~$Z(n_{M-1}, n_{M-2}, \dots, n_{M-m+1}, d_{m-1}, \dots, d_{M-2}, d_{M-1})$ by the kernel~~
 ~~$\{(-1)^{dr_{m-1}(n_{M-m} + n_{M-m+1}) + j(-1)^{di_{m-1}(n_{M-m} + n_{M-m+1})}\}$ and summing~~
10 ~~over $dr_{m-1}, di_{m-1}=0,1$ to yield the partially encoded symbol~~
~~set $Z(n_{M-1}, n_{M-1}, n_{M-2}, \dots, n_{M-m}, d_{m-1}, \dots, d_{M-2}, d_{M-1})$,~~
~~implement pass M of said fast encoding algorithm by~~
~~by multiplying $Z(n_{M-1}, n_{M-2}, \dots, n_2, n_1, d_{M-1})$ by the kernel~~
 ~~$\{(-1)^{dr_{M-1}(n_0 + n_1) + j(-1)^{di_{M-1}(n_0 + n_1)}\}$ and summing over~~
15 ~~$dr_{M-1}, di_{M-1}=0,1$ to yield the encoded symbol set~~
 ~~$Z(n_{M-1}, n_{M-1}, n_{M-2}, \dots, n_2, n_1, n_0)$,~~
~~reorder said encoded symbol set in memory in the ordered output~~
~~format $Z(n_0, n_1, \dots, n_{M-2}, n_{M-1})$ and store in said "D" and,~~
~~wherein said encoder in said CDMA transmitter reads said encoded~~
20 ~~symbol chip vector vector in said "D" and overlays said~~
~~vector with long and short pseudo-noise (PN) codes to~~
~~generate N chips of said hybrid Walsh encoded said~~
~~data symbol vector for subsequent processing and encoded~~
~~chip vector for transmission.~~

Claim 9. (currently amended) Wherein said hybrid Walsh
 codes in claim 5 have a fast decoding implementation algorithm,
 comprising the steps:

30 wherein the decoder strips off said pseudo-noise (PN) codes from
the received N chip encoded chip vector and writes the
resultant encoded chip vector $Z(n_0, n_1, \dots, n_{M-2}, n_{M-1})$ to
memory "A" wherein the binary addressing word takes address
values $n_0, n_1, \dots, n_{M-2}, n_{M-1}=0,1,2,\dots,N-1,$

wherein said fast decoding algorithm implemented in said decoder
uses said memory "A" for input and to support pass 1 and
memories "B","C" support passes 2, . . . ,M and memory "D"
stores the decoded data symbols from the rescaling and
5 reordering pass,

wherein pass 1 reads pairs of encoded chip symbols from "A"
corresponding to $n_0=0,1$ wherein the addresses of the chip
symbols for $n_0=0$ are $0,2,4,\dots,N-2$ and for $n_0=1$ are $1,3,\dots,N-1$ and for each pair of chip symbols pass 1 performs a
10 2-point hybrid Walsh inverse transform and sums the outputs
for each of the encoded data symbol index values $d_{M-1}=0,1$ and
writes the outputs to memory "B" using the addresses of the
respective chip symbols corresponding to $n_0=0,1$ and pass 1
processing generates the vector $Z(d_{M-1}, n_1, \dots, n_{M-1})$ in
15 "B",

wherein pass $m=2,3,\dots,M-1$ reads pairs of chip symbols from
"B","C","B", . . . and writes the outputs to "C","B","C", . . .
and chip symbol pairs for $n_{m-1}=0,1$ are read over address
blocks each of length 2^m , and starting with the first
20 address block the chip symbol $n_{m-1}=0$ is read for addresses
 $0,1,\dots,2^{(m-1)}$ and for $n_{m-1}=1$ the chip symbol addresses are
 $2^{(m-1)}+1, \dots, 2^m$ and for each pair of chip symbols pass
 m performs a 2-point hybrid Walsh inverse transform and
sums the outputs for each of the decoded data symbol binary
25 index values $(d_{M-m+1}, d_{M-m}=0,1)$ and writes the outputs to
addresses $(d_{M-m+1}, 0), (d_{M-m+1}, 1)$ corresponding to the addressed
of the input $n_{m-1}=0,1$ and this processing is repeated for
each of the address blocks and pass m processing
generates the data vector $Z(d_{M-1}, \dots, d_{M-m}, n_m, \dots, n_{M-1})$,

30 wherein pass M reads pairs of chip symbols from "B" or "C" and
writes the outputs to "C" or "B" and chip symbol pairs for
 $n_{M-1}=0,1$ are read over addresses $0,1,\dots,N/2-1$ for $n_{M-1}=0$
and over addresses $N/2,\dots,N-1$ for $n_{M-1}=1$ and for each
pair of chip symbols pass M performs a 2-point hybrid Walsh
35 inverse transform and sums the outputs for each of the

decoded data symbol binary index values $(d_0, d_1=0,1)$ and
 writes the outputs to addresses $(d_1,0), (d_1,1)$ corresponding
 to the addresses of the input $n_{M-1}=0,1$, and pass M processing
 generates the data symbol vector $Z(d_{M-1}, \dots, d_0)$,
 5 wherein a final pass scales the decoded data symbols by
 the N chip hybrid Walsh inverse transform scaling factor
 "1/2N" and re-orders the scaled encoded data symbols in
 memory "B" or "C" and stores the ordered output $Z(d_0, d_1, \dots$
 $\dots, d_{M-2}, d_{M-1})$ in memory "D", and
 10 wherein said decoder in said CDMA receiver reads said decoded
 data symbol vector in said "D" for further processing to
 recover information from the data symbols.
 15 ~~said decoder in said receiver strips off said PN codes from~~
~~— said received N chip encoded data symbol vector and outputs~~
~~said received hybrid Walsh encoded chip vector $Z(n_0, n_1, \dots$~~
 ~~$\dots, n_{M-2}, n_{M-1})$ for implementation of said fast decoding~~
 20 ~~algorithm,~~
~~said fast implementation algorithm in said decoder uses memory~~
~~— "E" for input and to support pass 1, memories "F", "G" to~~
~~— support passes 2,3, ..., M and re-ordering pass, and~~
~~— memory "H" for output,~~
 25 ~~write said $Z(n_0, n_1, \dots, n_{M-2}, n_{M-1})$ to said "E" wherein~~
 ~~$(n_0, n_1, \dots, n_{M-2}, n_{M-1})$ is the binary address,~~
~~pass 1 reads from said "E", performs pass 1, and writes the~~
~~— output to said "F",~~
~~implement pass 1 of said fast decoding algorithm by multiplying~~
 30 ~~said $Z(n_0, n_1, \dots, n_{M-2}, n_{M-1})$ by the kernel $\{(-1)^{n_0 d_{M-1}+j}(-$~~
 ~~$1)^{n_0 d_{M-1}}\}$ and summing over $n_0=0,1$ to yield the partially~~
~~decoded symbol set~~
 ~~$Z(d_{M-1}, n_1, \dots, n_{M-2}, n_{M-1})$,~~
~~write said output symbol set $Z(d_{M-1}, n_1, \dots, n_{M-2}, n_{M-1})$ to said~~
 35 ~~"F" wherein address index d_{M-1} replaces index n_0~~

~~pass 2 reads from said "F", performs pass 2, and writes the~~
~~output to said "G",~~
~~pass 3 reads from said "G", performs pass 3, and writes the~~
~~output to said "F",~~
5 ~~subsequent passes alternate in read/write from/to said "F" and~~
~~write/read to/from said "G",~~
~~implement passes m=2,3,...,M-1 of said fast decoding algorithm~~
~~by multiplying $Z(d_{M-1}, d_{M-2}, \dots, d_{M-m+1}, n_{m-1}, \dots, n_{M-2}, n_{M-1})$~~
~~by the kernel~~
10 ~~$\{(-1)^{n_{m-1}}(dr_{M-m} + dr_{M-m+1}) + j(-1)^{n_{m-1}}(di_{M-m} + di_{M-m+1})\}$ and summing~~
~~over $n_{m-1}=0,1$ to yield the partially decoded symbol set~~
 ~~$Z(d_{M-1}, d_{M-1}, d_{M-2}, \dots, d_{M-m}, n_{m-1}, \dots, n_{M-2}, n_{M-1})$,~~
~~implement pass M of said fast decoding algorithm by~~
~~by multiplying $Z(d_{M-1}, d_{M-2}, \dots, d_2, d_1, n_{M-1})$ by the kernel~~
15 ~~$\{(-1)^{n_{M-1}}(dr_0 + dr_1) + j(-1)^{n_{M-1}}(di_0 + di_1)\}$ and summing over~~
 ~~$n_{M-1}=0,1$ and rescaling by dividing by $2N$ to yield the~~
~~decoded symbol set~~
 ~~$Z(d_{M-1}, d_{M-1}, d_{M-2}, \dots, d_2, d_1, d_0)$,~~
~~reorder said decoded symbol set in the ordered output format~~
20 ~~$Z(d_0, d_1, \dots, d_{M-2}, d_{M-1})$ and store in said "H" and,~~
~~said decoder in said receiver reads said decoded symbol vector~~
~~in "D", re-orders the read data symbols to remove said~~
~~mapping onto said "A", and performs subsequent receive~~
~~signal processing to recover the information from the~~
25 ~~data symbols..~~

30